

# A Battery Lifespan-Aware Protocol for LPWAN

Sezana Fahmida<sup>†</sup>, Akshar Chavan<sup>‡</sup>, Venkata Prashant Modekurthy<sup>§</sup>, Abusayeed Saifullah<sup>†</sup>, Marco Brocanelli<sup>‡</sup>  
Wayne State University<sup>†</sup>, The Ohio State University<sup>‡</sup>, University of Nevada, Las Vegas<sup>§</sup>

**Abstract**—Energy harvesting sources, such as solar, wind, or vibration, combined with rechargeable batteries, are a promising way to power LPWAN (low-power wide-area network) devices to reduce the cost and frequency of redeploying single-use batteries. However, being oblivious to the usage of rechargeable batteries can severely reduce their *capacity to store energy*, which is also known as the *battery lifespan*. Existing energy-aware protocols mostly focus on network lifetime and pay little attention to maximizing the battery lifespan of network nodes while the latter can directly help reduce battery waste and enhance environmental sustainability. In this paper, we propose the first Media Access Control (MAC) protocol to maximize the minimum battery lifespan among all nodes in an LPWAN based on LoRa. Our approach differs from traditional objectives focusing on minimizing energy consumption or maximizing network lifetime as they may not necessarily maximize battery lifespan. The proposed MAC protocol leverages the concept of software-defined batteries to regulate the energy stored and consumed by each node's battery based on estimated energy requirements, green energy generation, and changes in data utility. To limit the degradation of battery capacity due to continuous charging/discharging, the underpinning idea is to determine an appropriate time for each transmission considering its impact on battery degradation while also minimizing the impact on data utility. Furthermore, the energy stored in each battery is limited to reduce *calendar aging*, the natural degradation of battery capacity over time. The proposed protocol is local, online, and asynchronous, and incurs low overhead. We evaluate our approach through experiments on a LoRa network and large-scale simulations in NS-3. The experiments show that the proposed MAC protocol improves battery lifespan by up to 69.7% and data utility by up to 39% in a current LoRa network while incurring a CPU utilization overhead of only 12% at each LoRa node.

## I. INTRODUCTION

Low-Power Wide-Area Network (LPWAN) has been gaining popularity as a preferred technology for many Internet-of-Things (IoT) applications due to its capability to support low-power (milliwatts), low data rate (kbps) communication over long distances (km) using narrowband (kHz) [1]. LPWANs are being used for a wide array of applications, including smart agriculture, environment monitoring, and smart utilities [2]. Many of these applications leverage thousands of sensors (nodes) sparsely distributed over hundreds of square kilometers in locations where line power is unavailable. Therefore, nodes use alternate sources of energy, such as single-use batteries, green energy sources, and rechargeable batteries. Typically, LPWAN devices remain operational longer than a single-use battery [3]. Moreover, in large-scale deployments, frequently redeploying batteries is tedious and expensive. With

the expected number of LPWAN devices growing to 1.3 billion by 2026 [4], 78 million batteries from IoT devices could be discarded every day [5], incurring severe environmental and financial burden [6], [7]. Therefore, alternative energy solutions that remain usable for a longer time are necessary to reduce the maintenance cost and carbon footprint of IoT applications.

Recently, energy harvesting sources, such as solar power [8], wind [9], and vibration [10], are being used to power the sensor nodes along with a small rechargeable battery to account for periods with no energy generation [11], [12]. Such hybrid energy sources present a sustainable, low-maintenance, and cost-efficient alternative to the larger single-use batteries. However, the ability to store energy, also known as the maximum capacity of a rechargeable battery reduces over time. Note that the maximum capacity of a battery is **different than** the State of Charge (SoC) of the battery. For example, a new/unused battery has 100% maximum capacity and can store the highest amount of energy, while a battery with decreased maximum capacity cannot store the same amount of energy, even with 100% SoC. The decrease in the maximum capacity is accelerated by a multitude of factors, including battery chemical properties, environmental and device temperature, number of charge-discharge cycles, average charge maintained in the battery within each cycle, and maximum discharge within each cycle (*depth of discharge*) [13]. A notable drop in the maximum capacity (usually 20%) renders the battery inoperable due to an exponential acceleration of capacity deterioration [14]. We define *Battery Lifespan* as the time elapsed from battery deployment until its maximum capacity falls below a predetermined amount, typically 80%. This work aims to maximize the battery lifespan for LPWANs.

Currently, LPWANs are not well-equipped to maximize the battery lifespan of the network. For example, LoRa [1], a leading LPWAN technology, relies on an ALOHA-like Media Access control (MAC) protocol, where a node tries to send a packet immediately after it is generated and does not consider any of the factors mentioned above, thereby ignoring battery lifespan. Existing work mostly focuses on maximizing network lifetime in LPWANs [15], which is **significantly different** than maximizing the **lifespan** of a rechargeable battery. For example, packet transmissions with energy-efficient parameters can significantly increase the depth of discharge of the battery if green energy is scarce at transmission time, thereby negatively impacting battery lifespan - a phenomenon known as *cycle aging*. On the other hand, when green energy is abundant and battery energy is rarely needed, the battery remains at a high SoC for long

This research was supported by the US National Science Foundation through grants CNS-1948365, CNS-2301757, CAREER- 2306486, CNS-2306745, CNS-2211640, and by ONR through grant N00014-23-1-2151.

time, which also deteriorates battery lifespan - a phenomenon known as *calendar aging*. Note that most of the network lifetime maximization approaches aim to maximize the operational time of a node for a single-use battery by keeping the SoC high, which may not be beneficial for rechargeable batteries' lifespan. Thus, novel communication strategies that focus on battery lifespan maximization are paramount for IoT.

In this paper, we propose the first MAC protocol for LPWAN that maximizes the minimum battery lifespan among all devices to increase the sustainability of the network. We specifically focus on LoRa as it is a dominant LPWAN technology that is commercially available across the world. However, our approach is applicable to most other LPWANs such as SigFox [16]. LPWANs are traditionally developed for time-insensitive applications requiring infrequent communications. The proposed MAC protocol exploits this application characteristic and leverages the concept of software-defined batteries [17] to meticulously regulate a node's battery energy consumption based on current and future energy consumption estimates. Specifically, it maximizes the lifespan of each battery by (a) delaying a packet transmission to a time when estimated green energy availability is high, thereby minimizing the energy drawn from the battery and cycle aging, and (b) limiting the energy stored in each battery to a fixed and predetermined threshold, thereby reducing calendar aging. Since typical LoRa applications are **not time-sensitive**, delaying packets by a small interval has **negligible impact** on their quality of service. However, long delays can decrease the usefulness of the data. The proposed approach attempts to balance the usefulness of the data and battery lifespan through a joint optimization. Furthermore, while limiting the energy stored in each battery may seem like a counter-intuitive approach, this is effective in increasing the battery lifespan by reducing calendar aging. Note that the proposed MAC protocol carefully selects transmission time to coincide with high green energy generation to limit battery usage. Therefore, the counter-intuitive approach of limiting the SoC to a predetermined threshold has a negligible effect on application requirements such as packet reception rate and data usefulness, while allowing for an increased battery lifespan.

Specifically, this paper makes the following contributions:

- We formulate the battery lifespan maximization problem for a clairvoyant central network manager. However, generating an optimal solution using the clairvoyant network manager has a high computational and synchronization overhead and is infeasible for low-power devices. Using the insights of this formulation, we develop a simple MAC protocol for a LoRa network based on a heuristic approach that enables each LoRa node to locally select a time for each transmission considering its impact on battery degradation and data utility based on the energy availability, the battery's SoC, and a utility function. The proposed protocol is **local**, **online**, and **asynchronous**, and incurs low overhead.
- We demonstrate the feasibility of our approach through small-scale testbed experiments on a LoRa network. In

addition, due to the impossibility of running 10-20 year-long experiments to measure battery lifespan on a real testbed, we present our key evaluation results based on NS-3 [18] simulations. Our approach provides up to 69.7% increase in battery lifespan with only 4% impact on average data utility and low overhead.

In the rest of the paper, Section II provides necessary background and system model. Section III describes the battery lifespan maximization problem and the proposed lightweight MAC. Section IV presents evaluation results. Section V and VI describe related work and conclusion, respectively.

## II. BACKGROUND AND MODEL

Here we describe the background for LoRa and battery degradation estimation in Section II-A and II-B, respectively. Finally, Section II-C describes the system model and objective.

### A. LoRa

LoRa is a pioneering LPWAN technology enabling information collection from thousands of sensors at a central node (gateway). It has over 600 known use cases and 225 million deployed devices. ABI research predicts communication on LoRa will account for 50% of all non-cellular LPWAN communications by 2026 [19]. Thus, in this paper, we focus on LoRa.

A typical LoRa network consists of three parts: gateway, end-devices, and a network server. *Nodes* or end-devices are sensors that communicate with the gateway via a single-hop wireless link. Multiple gateways communicate with the network server via a local LAN/the Internet. The network server in a LoRa network maintains the application requirement, security, and network parameters. A LoRa network divides the available spectrum into multiple *uplink* and *downlink* channels. Nodes send data to the gateway on the uplink channels and the gateway responds on the downlink channels.

LoRa uses a *Chirp Spread Spectrum (CSS)* based modulation. CSS provides robustness to interference and enables the reception of packets at a low/negative SNR by spreading the signal over the entire bandwidth. LoRa signal consists of multiple chirps, and each chirp contains encoded information [20]. The number of chirps present in a signal is controlled by *spreading factor (SF)*. LoRa supports SF in the range [7,12]. SF controls the data rate, the time on air, and the energy consumption of the transmission. A higher SF reduces the data rate, but increases the time on air and energy consumption.

Apart from SF, other configurable parameters for LoRa are carrier frequency, channel, bandwidth, and coding rate. In the US, LoRa operates in the unlicensed ISM band (902-928MHz) and is allocated 64 channels of 125kHz bandwidth and 8 channels of 500kHz bandwidth for uplink communication. For downlink communication, 8 channels of 500kHz bandwidth are used. LoRa also supports different levels of forward error correction (FEC), called *coding rates* in the range of  $\frac{4}{5}$  to  $\frac{4}{8}$ . A higher coding rate provides resilience against interference, but increases the size of each packet.

The MAC protocol used with the LoRa physical layer is called *LoRa Wide Area Network (LoRaWAN)*. In LoRaWAN, nodes transmit using pure ALOHA with pseudo-random channel hopping [21]. Upon transmission, nodes open two short receive windows to listen for packets from the gateway, referred to as class A mode of operation. LoRaWAN supports other modes of operation for downlink communication [21], which are not considered in the paper.

### B. Battery Degradation

Battery degradation models have been studied extensively, and proposing a new model is out of scope for this paper. Instead, the highly accurate and publicly available model proposed in [13] is used.

Typically, battery degradation of a node  $u$  happens due to two phenomena: *cycle aging* and *calendar aging*. *Calendar aging* refers to the natural degradation of battery capacity over time. It depends on the average internal temperature, the time elapsed since the manufacturing of the battery ( $\zeta$ ), and the average SoC across all charge-discharge cycles, calculated as the ratio of the current energy stored and the *original* maximum energy capacity of the battery. For a node  $u$ , the number of charge-discharge cycles is represented as  $N_u$  and the average internal temperature of a battery is represented as  $\bar{T}_u$ . We denote the average SoC of  $i^{\text{th}}$  charge-discharge cycle as  $\phi_u[i]$  and the set of average SoCs for *all* charge-discharge cycles as  $\phi_u$ , i.e.,  $\phi_u = \{\phi_u[i] | 1 \leq i \leq N_u\}$ . Considering  $\bar{\phi}_u$  to denote the average of all values in set  $\phi_u$ , the calendar aging is:

$$D_u^{cal}(\zeta, \bar{T}_u, \bar{\phi}_u) = k_1 \times \zeta e^{k_2(\bar{\phi}_u - k_3)} \times e^{\frac{k_4(\bar{T}_u - k_5)(273 + k_5)}{273 + \bar{T}_u}} \quad (1)$$

where,  $k_1, k_2, k_3, k_4, k_5$  are constants specific to the battery.

*Cycle aging* refers to the degradation caused by frequent charging and discharging of a battery. It depends on the number of charge-discharge cycles ( $N_u$ ), average SoC maintained within each cycle ( $\phi_u[i]$ ), maximum discharge within a cycle, and the average internal temperature of a battery within each cycle ( $\bar{T}_u$ ). We define *cycle discharge* as the difference between the maximum and minimum SoC within a cycle. For a node  $u$ , we represent the set of all cycle discharges as  $\delta_u$  and that of the  $i^{\text{th}}$  charge-discharge cycle as  $\delta_u[i]$ . Similarly,  $\eta_u[i]$  represents the cycle type of the  $i^{\text{th}}$  charge-discharge cycle of node  $u$ .  $N_u, \delta_u, \phi_u$ , and  $\eta_u$  can be computed from the battery capacity trace  $\Psi_u(0, \zeta)$  between time 0 to  $\zeta$  using the rainflow-counting algorithm [13]. Considering that the set of  $\eta_u[i]$  for all  $i$  is denoted by  $\eta_u$ , cycle aging of a battery is given by:

$$D_u^{cyc}(N_u, \delta_u, \phi_u, \bar{T}_u, \eta_u) = \sum_i^{N_u} \left( \eta_u[i] \times \delta_u[i] \times \phi_u[i] \times k_6 e^{\frac{k_4(\bar{T}_u - k_5)(273 + k_5)}{273 + \bar{T}_u}} \right) \quad (2)$$

where  $k_6$  is a constant specific to the battery.

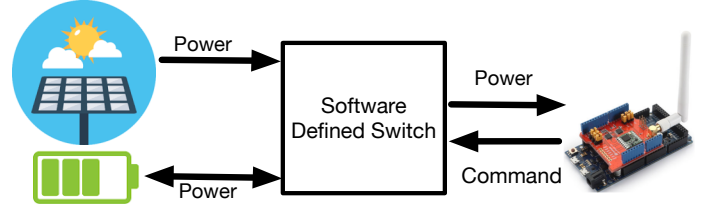


Fig. 1: System Model.

The linear degradation of node  $u$ 's battery, represented by  $D_u^L$ , is therefore calculated as:

$$D_u^L(\zeta, N_u, \delta_u, \phi_u, \bar{T}_u, \eta_u) = D_u^{cal} \left( \zeta, \bar{T}_u, \frac{\sum_i^{N_u} \phi_u[i]}{N} \right) + D_u^{cyc}(N_u, \delta_u, \phi_u, \bar{T}_u, \eta_u) \quad (3)$$

On the other hand, Solid Electrolyte Interphase (SEI) film formation in batteries introduces non-linearities in the model, which is not accounted for in Equation 3. Considering  $k$  is an estimated constant and  $\alpha_{sei}$  is the capacity lost due to SEI, the degradation of node  $u$ 's battery ( $D_u$ ) is an exponential function given by:

$$D_u(\zeta, N_u, \delta_u, \phi_u, \bar{T}_u, \eta_u) = 1 - \alpha_{sei} e^{-k \times D_u^L(\zeta, N_u, \delta_u, \phi_u, \bar{T}_u, \eta_u)} - (1 - \alpha_{sei}) e^{-D_u^L(\zeta, N_u, \delta_u, \phi_u, \bar{T}_u, \eta_u)} \quad (4)$$

### C. System Model and Objective

We consider a LoRa network consisting of thousands of nodes communicating with one or more gateways. Nodes are low-power devices that sleep most of the time and wake up periodically to sense the environment and report the collected information to the gateway. Typically, LoRa networks support applications that are not highly time-sensitive, such as data collection for smart farming and wildlife monitoring. Thus, the usefulness (or freshness) of the sampled data **decreases slowly** over time. Note, for most applications, a decrease in utility is acceptable while an *intermittent* operation (i.e., missed packets) is undesirable. Thus, the application expects data to be delivered within a *reasonable* delay.

As Figure 1 shows, nodes in the network have two energy sources: (1) a local green energy source (e.g., small solar panel) and (2) a rechargeable battery. The green energy source is sized to sustain packet transmission at peak generation. Furthermore, the battery is selected with a maximum capacity that is sufficient to maintain 24 hours of operation without recharging. The software-controlled switch regulates the power source of a node by controlling when the battery is charged and discharged. When the instantaneous power generated by the green energy source is greater than the node's power demand, the switch powers the node using only the green energy source, and the excess power generated can be used to charge the battery. Otherwise, the battery and the green energy source are used simultaneously to power the node. The instantaneous power generated by the green energy source varies with time. However, nodes can predict the amount of

green energy generated using a small neural network, trained at the gateway, as described in [22].

Typically, the maximum capacity stored in a battery decreases with every charge and discharge cycle. We refer to the *original maximum capacity* as the maximum energy that can be stored in a new battery. The drop in maximum capacity over time compared to the original maximum capacity is called *degradation* of a battery and is computed as the ratio of current maximum capacity and original maximum capacity. Upon reaching 20% degradation a batteries degradation increases exponentially, thus it is typically considered at the *End of Life* (EoL) [13] and is flagged for replacement. We define a node's *battery lifespan* as the time when the node's battery reaches EoL. The time and effort required for replacing just one battery is quite high and causes a severe disruption to day to day operations. Also, battery waste in such large-scale deployments can severely impact environmental sustainability. Thus, we define the *battery lifespan* of the *network*,  $\mathcal{L}$ , as the time interval starting from initial network operation until the *first* battery in the network reaches EoL. Note that, depending on the application requirement, this definition can easily be adapted to other percentages of nodes as well, such as the time until the first 10% of the batteries reach EoL.

Our goal is to minimize the maximum degradation by regulating the energy stored and consumed in each nodes' battery, which we refer to as the *battery lifespan maximization problem*. Note, we mainly focus on the transmission energy as it is generally higher than that of computation and sensing [20]. However, in Section IV-B we show our solution leads to a low resource consumption overhead.

### III. PROPOSED BATTERY LIFESPAN-AWARE MAC PROTOCOL

The battery lifespan maximization problem is formulated as a multi-objective mixed integer problem in Section III-A and uses the battery degradation model described in Section II-B. Since most battery models rely on the same parameters (e.g., average SoC, temperature), our formulation does not depend on any specific battery degradation model and any other model (with similar parameters) can be used. To the best of our knowledge, this is the first attempt to formulate the battery lifespan maximization problem in an LPWAN. However, the multi-objective formulation is hard to solve, and hence, we propose a **heuristic solution** in Section III-B.

#### A. Problem Formulation

The optimal solution for the battery lifespan maximization problem is a MAC protocol that regulates the SoC and the energy consumed during packet transmissions for each node, considering their current green energy availability and SoC. Since collisions increase communication energy consumption and decrease battery lifespan, the MAC should minimize packet collisions. Thus, we formulate a centralized problem considering a Time Division Multiple Access (TDMA) MAC protocol. Note that, TDMA-based MAC is **only used** for the optimal formulation and **not in our proposed solution**.

In the TDMA MAC, each time slot is long enough for a packet transmission and its acknowledgment (ACK) from the gateway. Since different SFs in LoRa can cause different transmission times, the time slot is long enough to transmit a packet on the highest spreading factor. Furthermore, LoRa gateways can receive multiple packets simultaneously on different channels. Therefore, each node transmits a packet on a unique combination of time slot and channel to avoid collisions.

Since the problem formulation is defined over a discrete-time model, we represent the degradation Equation 4 in the discrete-time model as  $D_u(\rho, N_u, \delta_u, \phi_u, \bar{T}_u, \eta_u)$ , where  $\rho$  is number of time slots in the interval  $[0, \zeta]$ . Furthermore, we can consider a discrete time trace of energy stored in the battery. Specifically, the discrete trace is generated after each time slot.

A solution to the battery lifespan maximization problem must identify the transmission time slot for each node in the network. Furthermore, it should also determine the percentage of green energy used for packet transmission and battery recharging within each time slot. Therefore, for node  $u$ , we introduce two sets of decision variables for time slots between 0 and  $\zeta$ ,  $X_u$  and  $Y_u$ . The binary decision variable  $x_u[t] \in X_u$  is equal to 1 if a packet transmission happens during time slot  $t$ , 0 otherwise.  $y_u[t] \in Y_u$  is a number in range  $[0, 1]$  that represents the fraction of green energy used in time slot  $t$  for powering the node, charge the battery, or both.

Considering a clairvoyant gateway having the energy generation  $E_u^g[t]$  of each node  $u$  during time slot  $t$ , the energy stored in the battery at time slot  $t$  ( $\psi_u[t]$ ) is:

$$\psi_u[t] = \psi_u[t-1] + y_u[t]E_u^g[t] - x_u[t]E_u^{tx} - (1-x_u[t])E_u^{sleep} \quad (5)$$

where,  $E_u^{sleep}$  and  $E_u^{tx}$  represent the energy consumed during sleep mode and energy consumed during packet transmission by node  $u$ , respectively. The green energy generation for each time slot can be predicted using existing algorithms as proposed in [22]. Given that each time slot is generally short, we safely assume it to be constant within each time slot. The energy consumption for a packet transmission in LoRa is [23]:

$$E_u^{tx} = P_u^{tx} \times L_u^{\text{symbols}} \times \frac{2SF_u}{BW_u} \quad (6)$$

where  $P_u^{tx}$  is the transmission power used by node  $u$ ,  $L_u^{\text{symbols}}$  is the number of symbols in a packet,  $SF_u$  and  $BW_u$  is the SF and the bandwidth used for transmission, respectively. The number of symbols in a packet is given by the following:

$$L_u^{\text{symbols}} = preamble_u + 4.25 + 8 + \max\left(\left\lceil \frac{8payload_u - 4SF_u + 24}{SF_u - 2DE_u} \right\rceil \frac{1}{CR_u}, 0\right) \quad (7)$$

where  $preamble_u$  represents the length of the preamble,  $payload_u$  represents the length of the payload, and  $DE_u$  is binary variable that is 1 if low-data rate optimization is enabled, 0 otherwise. The SF, coding rate, and DE used by each node in the network is known to the clairvoyant gateway.

Upon proper substitutions of (6) and (5) in (4), the degradation of a battery can be expressed in terms of a set

of decision variables  $X_u$ ,  $Y_u$ , and  $\rho$  as  $D_u(\rho, X_u, Y_u)$ . A simplistic approach would use an objective function that purely minimizes the maximum degradation  $D_u(\rho, X_u, Y_u)$ . However, such an optimization can overly delay a packet's transmission. For example, when there is no green energy source, the optimal solution can delay a packet's transmission until a green energy source is available to minimize cycle aging. To avoid such scenarios, we propose a bi-objective mixed integer non-linear problem where our objective is to minimize the maximum degradation and maximize the average utility (represented by  $\mu_u(X_u)$ ) of all packets. We define the *utility* of a packet as an indicator of the data usefulness at transmission time. Considering node  $u$  generates packets every  $\tau_u$  time slots, an example of utility of a single packet is a monotonically decreasing function from 1 to 0 as packet transmission time slot increases from the current packet's arrival to the next packet's arrival, and 0 after the arrival of the next packet. Note that the system designer can choose **different utility functions for different nodes**.

Since the average utility of a node is in the range  $[0, 1]$ , the optimization function on average utility can be expressed as a minimization of  $(1 - \mu_u(X_u))$ . Thus, the battery lifespan maximization problem of node  $u$  for  $\rho$  time slots is given by:

$$\begin{aligned} & \min D_u(\rho, X_u, Y_u) \\ & \min (1 - \mu_u(X_u)) \end{aligned}$$

The above optimization function maximizes the utility, but it does not imply transmissions of all packets. Therefore, we set a constraint that during the  $\rho$  time slots, node  $u$  should transmit all generated packets with the exception of the last one, i.e.,  $\lfloor \frac{\rho}{\tau_u} \rfloor < \sum_{t=0}^{\rho} x_u[t] \leq \lceil \frac{\rho}{\tau_u} \rceil$ . Here, the last packet will be considered for transmission during the subsequent runs of the optimization problem. Additionally, the objective function should avoid packet collisions, i.e., the number of transmissions made in the network at any time slot  $t$  is equal to the number of simultaneous receptions possible at the gateway. Considering the gateway can receive at most  $\omega$  simultaneous transmissions, the constraint is  $\sum_{\forall u} x_u[t] \leq \omega$ . Finally, for every time slot, the energy stored in node  $u$ 's battery is limited between 0 and the current maximum capacity ( $\psi_u^{\max}[t]$ ). Note that the current maximum capacity of the battery changes with degradation. Therefore, we set a constraint as  $0 \leq \psi_u[t] \leq \psi_u^{\max}[t] \forall u$  &  $\forall t$  s.t.  $0 \leq t \leq \rho$ . Thus, the *battery lifespan maximization problem* is given as:

$$\min \max_u D_u(\rho, X_u, Y_u) \quad (8)$$

$$\min \max_u (1 - \mu_u(X_u)) \quad (9)$$

$$\text{s.t. } \left\lfloor \frac{\rho}{\tau_u} \right\rfloor < \sum_{t=0}^{\rho} x(t) \leq \left\lceil \frac{\rho}{\tau_u} \right\rceil \quad \forall u \quad (10)$$

$$\sum_{\forall u} x_u[t] \leq \omega \quad \forall t \text{ s.t. } 0 \leq t \leq \rho \quad (11)$$

$$0 \leq \psi_u[t] \leq \psi_u^{\max}[t] \quad \forall u \text{ \& } \forall t \text{ s.t. } 0 \leq t \leq \rho \quad (12)$$

Since the energy estimation is accurate for  $\rho$  time slots, the above optimization can be executed at some arbitrary time slot  $\hat{t}$ . Note that, battery degradation starts at the beginning of time. Thus,  $X_u$  and  $Y_u$  are of size  $\hat{t} + \rho$ . However, only a subset of  $X_u$  and  $Y_u$  are considered as decision variables of the optimization function at time slot  $\hat{t}$ . Specifically, the subset of  $X_u$  and  $Y_u$  with values of  $t$  that satisfy  $\hat{t} \leq t \leq \hat{t} + \rho$  are decision variables. All other values are considered constants as they are calculated during previous iterations of optimization.

The above multi-objective mixed-integer problem can be hard to solve. The optimal algorithm generates decision variables for a short number of time slots, thereby has to be executed frequently. Furthermore, it requires a (impractical) clairvoyant network manager to estimate the green energy generated at each node. Realizing this in practice requires a lot of information to be collected from (and disseminated) to the nodes. In addition, the optimal solution leverages a TDMA-based MAC with no collisions to achieve low energy consumption at the node requiring very strong time synchronization, which causes high overhead and poor scalability. Thus, a centralized TDMA-scheduler is ill-suited for large-scale LoRa networks. For these reasons, we transform the above centralized problem into a **local problem** and propose an **online MAC** that solves it on **each sensor**.

## B. On-Sensor Approach

Here, we propose a local and online MAC for battery lifespan maximization. In the proposed MAC, each node locally decides its transmission time and green energy usage for the *current sampling period* by solving a *on-sensor* (simplified) version of the battery lifespan maximization problem. The on-sensor version of the battery lifespan maximization problem uses a *slotted-ALOHA* like MAC where each node locally divides its sensing period into several *forecast windows* for transmission and thereby decides the best forecast window to transmit. However, the nodes are **not time-synchronized**, eliminating overhead and making the proposed MAC highly scalable.

We aim to solve the battery lifespan maximization problem locally avoiding the overheads of information collection from and dissemination to the other nodes. There are three main challenges to do this: (1) the estimation of green energy generation (2) packet collisions arising from lack of synchronization between nodes and (3) computing and sharing the battery degradation information. We first explain how we address these challenges and then formulate the on-sensor problem.

**On-Sensor Green Energy Estimation.** The formulation in Section III-A relies on a clairvoyant gateway with accurate energy generation information of all nodes. This is difficult in practice due to huge overheads. Hence, in the on-sensor version energy generation of the local solar panel is estimated on each sensor. We define *forecast window* as the interval for which the node can accurately predict green energy generation.

Many existing photovoltaic (PV) power forecast solutions rely on real-time weather data such as wind speed and humidity for short (hours), medium (weeks), and long (months to year) term predictions [24]. These solutions cannot be

applied in our case study, since the sensors do not have expensive weather stations to sense weather data. Furthermore, the maximum transmission time for a 10-byte packet in LoRa is around 1.2 seconds, requiring a smaller forecast window. However, estimation of green energy generation for one packet transmission time is not feasible because (1) the green energy generation remains mostly constant across a couple of seconds, (2) a short forecast window length increases the overhead on the low-power nodes, and (3) packet collisions in one forecast window can cause a cascade effect of collisions in subsequent forecast windows due to retransmissions, causing a significant overhead. Thus, we consider the forecast window length to be long enough for 8 retransmissions (maximum allowed by LoRa) on the highest SF and to be aligned with the minimum granularity of green energy forecast feasible at the node.

The models proposed in [22] use only locally-available variables (e.g., recent PV power generation) to perform very-short term forecasts (1 to 30 min) and thus may be used at the low power LoRa nodes. As it is out of scope for us to propose new green energy prediction models, we assume these existing models are trained offline and deployed on each sensor. Since LoRaWAN Class A devices open two long receive windows after each transmission, it takes around 40 seconds to finish 8 retransmissions (including receive windows) of a 10 byte packet at SF 10. Considering the minimum granularity of very-short term forecast, we suggest a forecast window length of 1-2 minutes. However, the length of the forecast window may be set to an appropriate value depending on the green energy source, the granularity of green energy prediction model, and other network parameters.

**Compensating for packet collisions.** The centralized formulation in Section III-A is based on a TDMA scheduler, with no packet collisions. However, in the on-sensor approach, the nodes are not time synchronized, and transmission time selection is local to a node. Thus, multiple nodes may transmit in the same forecast window, resulting in packet collisions and retransmissions. If too many nodes transmit in the same forecast window, some packets may fail to reach the gateway. Furthermore, a high number of retransmissions results in high energy consumption, which may hamper the lifespan of a node. To solve this, we exploit the increased energy consumption as an indicator to refrain from selecting the crowded forecast windows through energy usage estimation.

The power and computational constraints in LoRa nodes hinder the adoption of complex estimation methods. Thus, we rely on simple but effective approaches to estimate energy usage. The energy usage in a forecast window depends on (1) the energy consumed per transmission, and (2) the number of retransmissions. In LoRa, the nodes can change their transmission parameters dynamically as governed by the underlying MAC layer or the network server. Parameters may also change due to channel conditions. Directly using the current transmission parameters to estimate the energy usage in the next sampling period may result in high variance, which is often not desirable. Thus, we use *Exponentially Weighted Moving Average (EWMA)*, which is often used to describe

time-series data. EWMA is designed to give more or less importance to newer data compared to older one. Thus, the transmission energy estimate is:

$$\mathbf{e}_u^{tx}[p] = \beta \cdot E_u^{tx}[p-1] + (1-\beta) \cdot \mathbf{e}_u^{tx}[p-1] \quad (13)$$

Here,  $\mathbf{e}_u^{tx}[p]$  is the transmission energy estimate in the current sampling period  $p$ ,  $E_u^{tx}[p-1]$  is the actual transmission energy consumption in the previous period and  $\beta$  is the importance weight decided by the network manager.

Accurate estimation of retransmission for a packet is difficult as it depends on unpredictable channel conditions. However, our main objective is not a highly accurate retransmission estimation strategy considering the channel dynamics. Our goal is to provide a reasonable estimate of retransmissions for the choice of forecast windows. We consider that if a node experiences a high number of retransmission in the previous forecast window, there is a high probability that choosing the same forecast window again would result in a high number of retransmissions. Thus, we calculate the probability of retransmissions in a forecast window based on historical data. As more data is gathered over time, we expect the estimation to be more accurate. In our approach, we assume that the transmissions on one forecast window have a negligible effect on transmissions in other forecast windows. Thus, we estimate the probability of retransmissions on each forecast window independently. Specifically, the probability of encountering  $r$  retransmissions in forecast window  $t$  is:

$$\mathbf{P}(r|t) = \begin{cases} \frac{I_{r,t}}{S_t} & \text{if } r = 0 \\ \mathbf{P}(r-1|t) + \frac{I_{r,t}}{S_t} & \text{if } r > 0 \end{cases} \quad (14)$$

where  $I_{r,t}$  and  $S_t$  are the number of times  $r$  retransmissions are observed in forecast window  $t$  and forecast window  $t$  is selected for transmission, respectively. We leave the analysis of physical channel conditions across forecast windows as future work. *Note, any energy usage estimation technique with reasonable overhead can be used in our approach.*

**Computing Battery Degradation.** The solution in Section III-A relies on the global knowledge of battery degradation of all nodes in the network. However, this is not practical for the on-sensor version for two reasons. First, battery degradation is calculated through a computation-intensive rainfall algorithm, which may not be feasible for low-power nodes. Second, a typical LPWAN network consists of thousands of nodes, and sharing battery degradation can be time and energy consuming. To this end, we propose to compute nodes' battery degradation at the gateway. However, the gateway requires the SoC trace to compute battery degradation. Next, we quantify the overhead of sharing the SoC trace.

**Overhead of sharing battery trace.** We observe that the SoC ( $\psi_u[t]$ ) at the forecast window ( $t$ ) when the battery transitions from charging to discharging and vice-versa are sufficient to generate the entire trace. Therefore, the nodes append forecast window  $t$  and  $\psi_u[t]$  at each battery transition time during the last period to the packet. The size of  $\psi_u[t]$  depends on the number of times charging (or discharging) occurs within a

sampling period. Note that, energy consumed in sleep mode remains similar for all forecast windows, therefore a node significantly discharges in only one forecast window for packet transmission. However, due to green energy variation, the node may recharge multiple times during a sampling period. We observe that only the last time of recharge is sufficient for calculating battery degradation. Thus, we include data for only two forecast windows, namely the forecast window where the node discharges and the last forecast window of recharge. A hardware interrupt is used to record the last forecast window of recharge. Thus, the information generated is minimal and can be appended to the subsequent packet with low overhead. Specifically, packet size increases by 4 bytes ( $2 \times 2$  bytes for  $t$  and  $\psi_u[t]$ ) resulting in 41 ms of additional airtime using SF 10 and BW 125 kHz.

**Disseminating battery degradation.** The gateway maintains the trace of energy stored in the battery and computes the battery degradation of each node. The gateway also computes the normalized battery degradation of each node. The normalized battery degradation of node  $u$  in the network is calculated as  $w_u = \frac{D_u}{D_{max}}$ , where  $D_{max}$  is the maximum degradation in the network. We propose to disseminate the normalized degradation to each node in the network. The normalized degradation can be used as the importance of the degradation over packet utility. This lets nodes with low degradation choose an early forecast window to maximize utility, even if it slightly increases battery degradation. On the other hand, the nodes with high degradation select forecast windows with lower collisions and higher green energy source, which reduces their degradation rate. *Each node only needs to know its own normalized degradation.* Thus, this strategy allows sensors to *indirectly* coordinate with each other and increase the network battery lifespan *without synchronization*.

**Overhead of sharing degradation.** We observe that the per-day change in the degradation of a typical battery is extremely small (between 0.001 and 0.0001). Therefore, the normalized battery degradation can be disseminated infrequently, i.e., once a day or month. Furthermore, the change in normalized degradation is minimal. Thus, the approach can tolerate delays in disseminating normalized battery degradation information, and the nodes do not need to receive the battery degradation information at the exact time. We propose to leverage on the above observation and disseminate the normalized battery degradation information once a day using piggy-backed ACKs. This approach adds minimal overhead to each acknowledgment packet. Specifically, the packet size increases by 1 byte, as the gateway only adds the normalized degradation of the node for which the ACK is intended. **On-sensor Formulation.** Here, we present the on-sensor formulation of the battery lifespan maximization problem where the decision variable  $X_u$  and  $Y_u$  pertain to one node only, i.e. for node  $u$ ,  $X_u = x_u[t]$  and  $Y_u = y_u[t]$ . To simplify the problem space, we only consider the decision variables in the current sampling period. However, as the battery degradation calculation is computationally expensive for the low-power nodes, we cannot directly use the same objective as Section III-A. Instead, we need a new

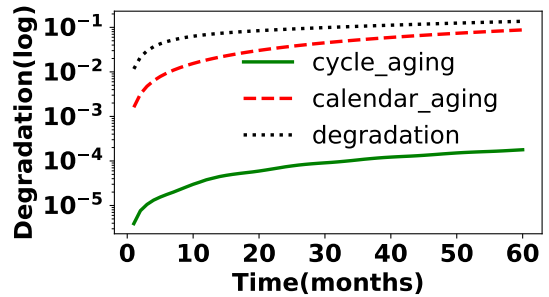


Fig. 2: Battery degradation.

objective that approximates the impact of packet transmission ( $x_u[t]$ ) and green energy usage ( $y_u[t]$ ). Both decision variables control the SoC of the battery. Specifically,  $x_u[t]$  controls the discharge by selecting the forecast window for transmission, and  $y_u[t]$  controls the recharge by green energy.

In Figure 2, we show the battery degradation (using Equation 4) of a regular LoRa node over the timespan of 5 years in a network consisting of 100 nodes where each node used a random transmission interval within [16,60] minutes. As can be seen from the figure, degradation due to calendar aging is significantly higher than degradation due to cycle aging. As such, calendar aging becomes the dominant factor in determining the final degradation. While it may seem like a high SoC will be beneficial to the battery lifespan, we observe that this assumption does not hold in case of a LoRa node due to the high impact of calendar aging. According to Equation (1), degradation monotonically *increases* with mean SoC of all charge-discharge cycles. Thus, reducing the mean SoC limits the calendar aging of a battery. To simplify the decision space for the local problem we select the  $y_u[t]$  that limits the maximum energy stored in the battery to a predefined threshold  $\theta$ , i.e., for a node  $u$ ,  $\psi_u[t] < \theta \quad \forall t$ . The network manager may configure  $\theta$  considering the application requirement.

Next, we aim to reduce the cycle-aging component of the overall degradation through a novel battery-lifespan aware MAC protocol. As we have already limited the maximum SoC by  $\theta$ , we now aim to transmit the packet in a forecast window that limits the discharge of the battery as much as possible. This results in a lower number of charge-discharge cycles and low cycle discharge, thereby resulting in a low cycle aging according to Equation (2). We approximate the impact of transmitting in a forecast window  $t$  on degradation through *Degradation Impact Factor* ( $DIF_u[t]$ ) as:

$$DIF_u[t] = \frac{\max(\mathbf{e}_u^{tx}, E_u^g[t]) - E_u^g[t]}{E_{max}^{tx}} \quad (15)$$

where,  $E_{max}^{tx}$ ,  $\mathbf{e}_u^{tx}$ , and  $E_u^g[t]$  are the maximum energy consumption for a transmission, *estimated* energy consumption, and energy generation for forecast window  $t$ , respectively.  $DIF_u[t]$  is a real number in the range [0, 1] that indicates the impact of transmitting in a forecast window  $t$  on degradation. In Equation (15), the SoC may decrease if  $\mathbf{e}_u^{tx} > E_u^g[t]$ , which in turn increases the degradation due to cycle aging. In this

case  $DIF_u[t] > 0$ . On the other hand, if  $e_u^{tx} \leq E_u^g[t]$ , the SoC either increases or remains unchanged, both of which have negligible impact on cycle aging and thus  $DIF_u[t] = 0$ .

Solely relying on  $DIF_u[t]$  for deciding the forecast window for transmission may introduce large delays in packet reception. To this end, like the centralized approach in Section III-A, the on-sensor objective also takes packet utility into account. However, the multi-objective optimization problem in Section III-A is difficult to solve for the low-power LoRa nodes. Thus, we approximate the multi-objective optimization function as a weighted average of utility  $\mu_u[t]$  and  $DIF_u[t]$ . The utility  $\mu_u[t]$  may be expressed with any monotonically decreasing function of time such as:

$$\mu_u(X_u) = \frac{\tau_u - \sum_{t=0}^{\tau_u} t * x_u[t]}{\tau_u} \quad (16)$$

Here, only one packet transmission of node  $u$  is considered and  $\tau_u - \sum_{t=0}^{\tau_u} t * x_u[t]$  is the difference between the arrival of the next packet and transmission of the current packet. *Note that, our approach is not specific to the function in Equation 16 and can be used with any utility function.*

The normalized degradation of a node  $u$  from the gateway is represented by  $w_u$ . The weight  $w_b \in [0, 1]$ , chosen by the network manager depending on the application requirement, denotes the importance of degradation over the utility of a packet. Considering  $\mathcal{T}$  is the set of forecast windows within a sampling period, the objective function for the on-sensor approximation of battery lifespan maximization problem is:

$$\min \sum_{t \in \mathcal{T}} x_u[t] \cdot ((1 - \mu_u(X_u)) + w_u \cdot DIF_u[t] \cdot w_b) \quad (17)$$

One goal of Equation (17) is to maximize the minimum battery lifespan of the network. We realize this goal through the normalized degradation  $w_u$ . For a node with a higher degraded battery, the impact of  $DIF_u[t]$  is higher on the objective, thus the node will try to preserve its battery lifespan more conservatively than other nodes. However, nodes with newer batteries will have lower impact from the  $DIF_u[t]$ , so they will prioritize utility of the packet. This inherently results in a fair distribution of degradation in the network. The on-sensor and online approximation of the battery lifespan maximization problem is presented below:

$$\min_t \sum_{t \in \mathcal{T}} x_u[t] \cdot ((1 - \mu_u(X_u)) + w_u \cdot DIF_u[t] \cdot w_b) \quad (18)$$

$$s.t. \quad \sum_{t \in \mathcal{T}} x_u[t] = 1 \quad (19)$$

$$\psi_u[t-1] + e_u^g[t] \geq x_u[t] \cdot e_u^{tx}[t] \quad (20)$$

$$\psi_u[t] = \min(\theta, \psi_u[t-1] + E_u^g[t] - x_u[t]E_u^{tx}) \quad (21)$$

where  $\mathcal{T}$  is the set of forecast windows in the current sampling period. Equation (19) ensures exactly one forecast window is chosen in a period, Equation (20) ensures that the battery can sustain the transmissions in the chosen forecast window, while Equation (21) updates the current energy stored in the battery with the minimum of Equation (5) and  $\theta$ . The above problem where a forecast window is selected for transmitting a packet

---

### Algorithm 1 On-Sensor Forecast Window Selection

---

**Input:**  $\mathcal{T}$ , energy level ( $\psi$ ),  $w_u$ ,  $w_b$ ,  $\{E_u^g[t] | t \in \mathcal{T}\}$ ,  $\{e_u^{tx}[t] | t \in \mathcal{T}\}$

**Output:** SUCCESS/FAIL, Decision variable set ( $X_u = \{x_u[t] | t \in \mathcal{T}\}$ )

```

1:  $X_u \leftarrow \phi$ 
2: for  $t \in \mathcal{T}$  do
3:    $x_u[t] \leftarrow 0$ 
4:    $\gamma_t \leftarrow \mu_u[t] + w_u \cdot DIF_u[t] \cdot w_b$ 
5:    $X_u \leftarrow X_u \cup x_u[t]$ 
6: end for
7:  $\mathcal{T}' \leftarrow$  Sort  $\mathcal{T}$  in non-decreasing order of  $\gamma_t$ 
8:  $E_0 \leftarrow \psi$ 
9: for  $t \in \mathcal{T}$  do
10:   $E[t] \leftarrow E[t-1] + E_u^g[t]$ 
11: end for
12: for  $t \in \mathcal{T}'$  do
13:  if  $E[t] - e_u^{tx}[t] > 0$  then
14:     $x_t \leftarrow 1$ 
15:    Return SUCCESS,  $X_u$ 
16:  end if
17: end for
18: Return FAIL,  $X_u$ 

```

---

from a node in the current period can be solved in polynomial time, as  $w_u$  is constant for the day.

**On-Sensor Forecast Window Selection.** The pseudo-code to select a forecast window at a node is shown in Algorithm 1. It takes as input the set of forecast windows ( $\mathcal{T}$ ), current energy level of the battery ( $\psi$ ), degradation weight ( $w$ ), sets of estimated harvested energy ( $\{E_u^g[t] | t \in \mathcal{T}\}$ ) and estimated energy consumption ( $\{e_u^{tx}[t] | t \in \mathcal{T}\}$ ). The loop in Line 2, initializes the decision variable set  $X_u$  and evaluates the objective function for each forecast window,  $\gamma_t$ .  $\mathcal{T}$  is then sorted in non-decreasing order of  $\gamma_t$  and the result is stored in  $\mathcal{T}'$ . After calculating the estimated SoC  $E[t]$  for each forecast window  $t$  in Line 9, the loop in Line 12 traverses  $\mathcal{T}'$  and the forecast window with the lowest objective value that satisfies the constraint in Equation (20) is returned. Otherwise, the algorithm returns FAIL and the packet is dropped. This may be the result of a low  $\theta$ , which was not sufficient to sustain the network in the intervals without energy generation (e.g., night hours). Alternatively, there was no energy generation for an extended period of time. Thus, it is understandable that the node may not be able to deliver all packets. The time complexity for this algorithm is  $O(|\mathcal{T}| \log |\mathcal{T}|)$ . For a node with 10 Min period and 1 Min forecast window,  $|\mathcal{T}| = 10$ .

**Network dynamics and channel access.** The transmission time of a packet within a forecast window is configurable and should be governed by the application requirement and the utility function. For example, if the utility of the packet does not change significantly between the interval  $[0, L]$ , where  $L <$  length of the forecast window, then the node may choose a transmission time randomly within  $[0, L]$ . This reduces the

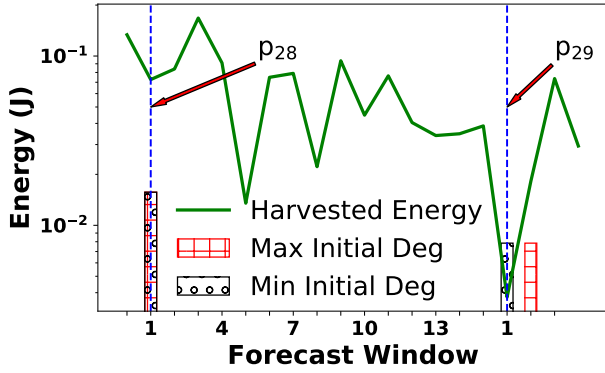


Fig. 3: Degradation influence.

chance of collisions in the same forecast window. The channel and SF selection are similar to LoRaWAN in our protocol. Finally, when a new node joins the network with an unused battery, its normalized degradation is 0. Thus, it can run Algorithm 1 without any communication with the gateway.

#### IV. EVALUATION

Here, we evaluate our approach to maximize the minimum battery lifespan of a network. Note that evaluating network battery lifespan through **physical experiments** is **difficult** as it usually takes **10-20 years to reach 20% battery degradation**. The measurement precision is also not enough for capturing the change in degradation in a feasible time. Thus, we run long (e.g., 15 years), large-scale experiments using NS-3 [18]. To show the feasibility of our approach, we also complement the large-scale results with short (e.g., 24 hours), small-scale results from the **physical testbed** in our labs.

##### A. Large-Scale Results

1) *Setup*: We implemented Algorithm 1 and the battery degradation model in [13] on top of the LoRaWAN NS-3 module [25] in our simulation.

We used up to 500 nodes and a single gateway. Nodes' locations were selected randomly with a maximum distance from the gateway of 5 km, simulating a dense deployment. Nodes use a sampling period randomly chosen from [16, 60] Min, forecast window of 1 min, and  $w_b = 1$ . We consider the battery to be insulated and thereby use a fixed internal temperature of 25° Celsius. We used the year-long green energy trace from a real solar panel [26]. The solar trace was scaled to generate, at peak power, enough energy to support two transmissions. Since the nodes are spread over a wide area, random variations were introduced in the solar power trace to emulate cloud cover and shades occurring over the deployment area.

2) *Baseline and metrics*: To our knowledge, no other work maximizes the minimum battery lifespan of an LPWAN. Thus we compare our approach with regular LoRa. Note, other MAC protocols that do not take the green energy availability and current battery capacity into account when transmitting packets will perform similarly to LoRaWAN

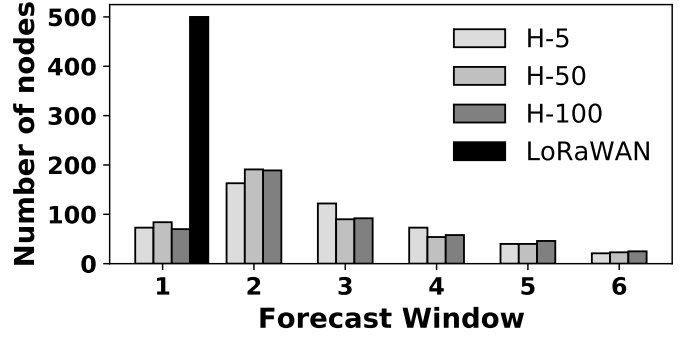


Fig. 4: Forecast window selection

in battery lifespan. The metrics used in our simulations are **avg retransmissions (RETX)** attempts per packet, total **transmission (TX) energy** (Equation (6)) for the network operation time, **battery degradation** (Equation (4)), **Packet Reception Rate (PRR)**, the ratio of number of ACKs received to number of packets generated), **avg utility** (Equation (16)) per packet, **avg latency** for each packet (interval between packet generation and reception of ACK). We penalize failed packets with a latency of the sampling period.

3) *Effect of weight  $w_u$  on forecast window selection*: Here, we use the same solar power trace and sampling period for the highest and lowest degraded nodes in a 100-node network. Figure 3 shows the forecast window selection of two nodes with the highest and lowest degraded battery. In sampling period  $p_{28}$  the energy generation is higher than the required energy for packet transmission. Thus, both nodes select forecast window 1 with high utility. However, in sampling period  $p_{29}$  when the harvested energy is lower than the energy required for transmission, the highest degraded node selects forecast window 2 to minimize cycle aging while the lowest degraded node still selects forecast window 1 to maximize utility. Thus, our approach reduces the network traffic for nodes with degraded batteries using the weight  $w_u$ .

4) *Effect of maximum SoC threshold ( $\theta$ )*: We run a simulation of 5 years with 500 nodes with unused batteries and vary  $\theta$  from 5% to 100%. In all results, H-5, H-50, and H-100 denote Algorithm 1 with  $\theta = 0.05, 0.5, 1$ , respectively.

**Forecast window selection.** Figure 4 shows the forecast window selection of the nodes in the network. Each bar indicates the number of nodes that transmitted the majority of their packets in the forecast window shown in x-axis. LoRaWAN always selects the first forecast window. However, for H-50, 38% of the nodes select forecast window 2 while 17% of the nodes select forecast window 1. Also, most of the nodes transmit within the first 4 forecast windows for all  $\theta$ . Thus, our algorithm distributes the nodes across different forecast windows, regardless of  $\theta$ .

**Transmission energy and battery degradation.** We show the battery degradation, avg RETXs and TX energy consumption of the network over 5 years in Figure 5. In Figure 5a, the avg RETX attempts for all  $\theta$  is lower than LoRaWAN. In

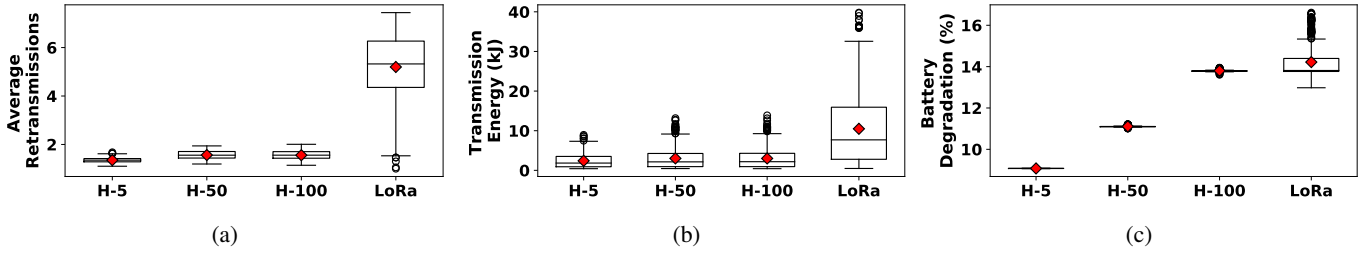


Fig. 5: (a) Avg Tx Attempts, (b) Tx Energy, and (c) Battery degradation under varying charging threshold  $\theta$ .

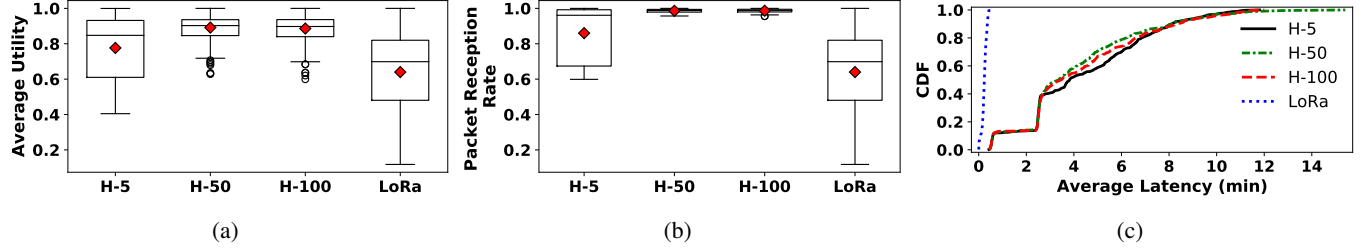


Fig. 6: (a) Avg Utility (b) PRR (c) Avg Latency under varying charging threshold  $\theta$

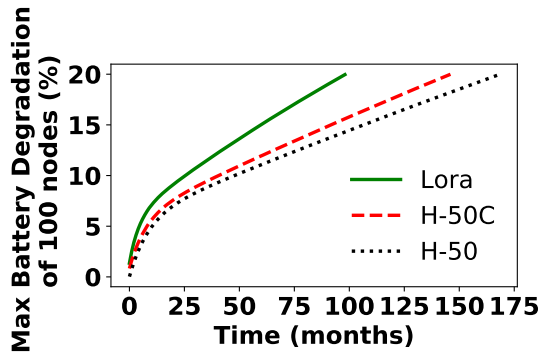


Fig. 7: Max degradation (%) of the nodes.

fact, H-50 reduces the RETX attempts by 69.9% compared to LoRa. We observe a similar trend in TX energy consumption in Figure 5b as our approach reduces collisions by selecting different forecast windows for different nodes (see Figure 4). In Figure 5c the mean degradation of H-100 is similar to LoRaWAN with a lower variance (indicated by the lower number of outliers). Note, H-100 uses  $\theta = 1$ , which does not minimize calendar aging. However, H-50 minimizes the mean degradation of the network by 21.88% and the variance of the degradation by 91.5% compared to LoRaWAN. While H-5 results in the minimum battery degradation, it may impact network performance as discussed next.

**Network and data performance.** Figure 6 shows the avg utility, PRR, and avg latency of the nodes over 5 years. In Figure 6a and 6b, the avg utility and PRR for LoRaWAN vary in a wide range (the lowest PRR is 63.9%) due to the Pure ALOHA MAC. The PRR for H-5 is lower than H-50 and H-100, as most of the nodes deplete batteries due to low  $\theta$ . As expected, 100% of the nodes using LoRaWAN have a

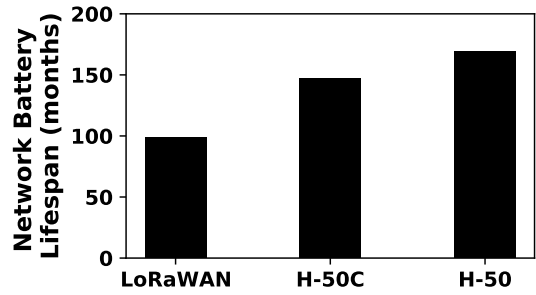


Fig. 8: Network Battery lifespan

latency lower than 35.11 s in Figure 6c. This is due to the Pure ALOHA MAC used by default LoRaWAN, which tries to transmit packets immediately. The avg latency for H-50 is 247.17 s. This is expected as we use  $w_b = 1$  and thus nodes may transmit in a later forecast window to preserve battery lifespan. **While this latency increase may seem high, it does not impact the avg utility in Figure 6a.** In fact, compared to LoRaWAN, H-50 *improves* the avg utility and PRR by 39% and 54%, respectively. Thus, our approach results in a high PRR providing higher utility than LoRaWAN. Note, the latency is configurable by the weight  $w_b$ . Low values of  $w_b$  result in a lower latency at the cost of a lower battery lifespan. **Battery lifespan.** Here, we evaluate battery lifespan by simulating the network until the first node reaches 20% degradation. To reduce the execution time, we use 100 nodes and evaluate H-50 and another variant of our approach, H-50C, which only uses  $\theta = 0.5$  without using our forecast window selection algorithm. Figure 7 shows the maximum degradation in the network at the end of every month for this simulation. The rate of degradation for LoRaWAN is higher in Figure 7 than H-50 and H-50C, resulting in a battery lifespan of 2980 days

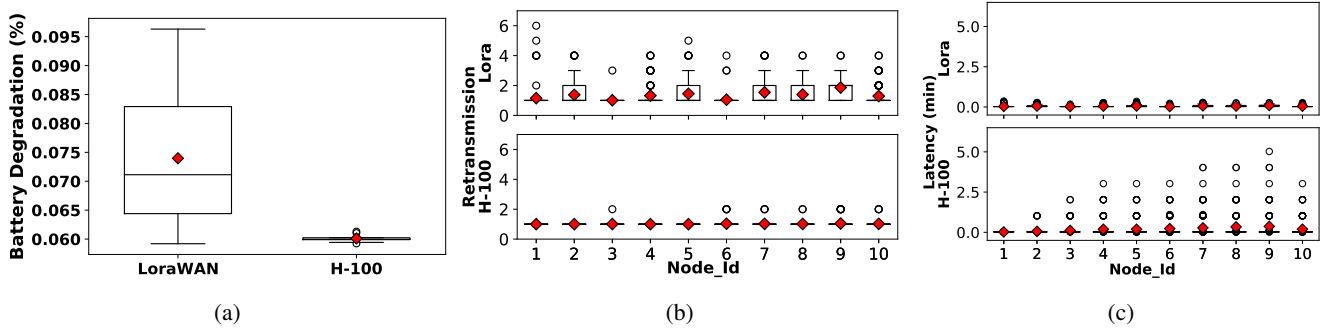


Fig. 9: (a) Battery Degradation, (b) Retransmission, and (c) Latency of 10 real nodes with H-100 and LoRaWAN.

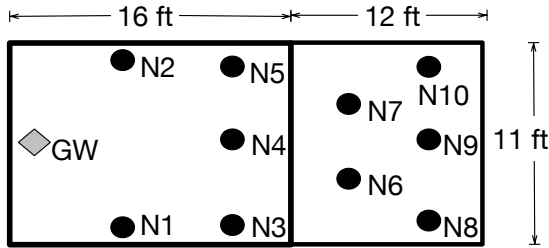


Fig. 10: Deployment map

(8.1 years) in Figure 8, which is 41.09% lower compared to H-50. Therefore, LoRaWAN is not able to sustain the battery capacity for the planned deployment time (typically 10+ years [3]) and requires battery replacement after 8 years, incurring high costs and wastage of natural resources. H-50 has a lifespan of 13.86 years providing additional 4 years of lifespan compared to LoRaWAN. *Note that, even if the planned deployment time is less than 13.86 years, the batteries can be reused in another deployment, thereby promoting sustainability and reducing cost.*

### B. Small-Scale Testbed Results

Here, we use 10 Dragino SX1276 LoRa transceiver HAT [27] on Raspberry pi 3 [28], as LoRa nodes and implement Algorithm 1 on top of LMIC 1.6 LoRa/LoRaWAN [29] library. All nodes use 10 min sampling period and forecast window of 1 min. We use the RAK2245 HAT [30] on Raspberry Pi 3 as the gateway with a local chirpstack network server [31]. To emulate a larger network, we use one 125 kHz channel in US 902 MHz band with SF 10. The experiment was conducted for 24 hours using a random day from the year-long energy trace [26] in an indoor environment (Figure 10). Note that the static energy consumption of the raspberry Pi is significantly high, which would rapidly drain a battery leading to high degradation values that are impractical for a LoRa network. Since our goal is to show the effect of LoRa Tx on battery degradation, we emulate the battery by a local variable on every node and update it after each forecast window using Equation (5). We evaluate our approach with  $\theta = 0.5$  and compare it with LoRaWAN.

1) *Results:* PRR is 100% for both approaches since LoRaWAN can easily maintain PRR for a small network with RETXs. Figure 9 shows the battery degradation, avg RETXs, and latency of the 10 nodes. In Figure 9a, the variance in battery degradation for LoRa is 99.7% higher compared to our approach, which ensures a fair distribution of degradation. The mean degradation in 24 hours is not sufficient for a significant improvement in overall non-linear degradation. However, even in 24 hours the cycle aging (Equation (2)) is 80% lower in our approach than LoRaWAN. Thus, in the long run, our approach will gain significant improvement, as verified by the results in Section IV-A. In Figure 9b, H-100 has lower number of RETXs than LoRa as we select different forecast window for transmission based on the green energy availability. In Figure 9c, the latency for LoRa is lower than

	LoRaWAN	H-100
Avg CPU util (%)	19.9	22.4
Memory util (%)	0.067	0.07
Exe size (kB)	56	60
USS (kB)	242	248

TABLE I: System overhead

our approach, showing a similar trend as the large-scale results.

We used the `psutil` python package [32] to monitor the system utilization shown in Table I on a node for 30 min for regular LoRaWAN and the proposed MAC. The avg CPU and memory utilization for our approach is 12.56% and 5.73% higher, respectively. The Unique Set Size (USS) (memory that is freed if the process terminates) increased by 2.61% in our approach, while the executable code size is only 7.14% higher, showing a small overhead.

## V. RELATED WORK

Initial works on LoRa have focused on empirical evaluation, while many works have also studied network performance enhancement such as throughput, latency, reliability, or scalability and coverage [33]–[35]. However, none of these works consider the problem of maximizing the minimum battery lifespan of the network. Network lifetime maximization approaches [15], [20] do not maximize battery lifespan, since reducing the

TX energy consumption may result in a high SoC being maintained in the battery. This can deteriorate battery degradation due to calendar aging. Also, transmitting a packet when green energy is scarce may result in a high depth of discharge even with the most energy-efficient transmission parameters. Thus, these approaches cannot be directly applied to prolong battery lifespan. Reducing battery degradation in Electric Vehicles, cell phones, and real-time task scheduling [36]–[38] focus on regulating computation on the processor to minimize degradation and cannot be adapted to LPWANs. A recent work [39] has proposed a hybrid power source with supercapacitors to minimize the degradation of the battery for ultra-low traffic scenarios (one packet every 3 hours) in LPWAN. Such hardware models cannot support transmissions in periods with no energy availability and thus do not eliminate the necessity of batteries for periodic data collection. Therefore, the software-defined battery-based approach proposed here is still applicable to regulate the charging/discharging of the platform. We leave the study of setups considering supercapacitors as future work. In contrast, we propose a novel MAC protocol that regulates the energy stored and consumed by the battery based on estimated current and future energy requirements of a node. *To the best of our knowledge, this is the first MAC protocol that regulates packet transmissions to maximize battery lifespan.*

## VI. CONCLUSION

Maximizing the battery lifespan of an LPWAN is crucial for ensuring sustainability of IoT applications. This paper proposed a lightweight MAC that preserves the battery lifespan of LoRa networks, a leading LPWAN technology, with minimal impact on utility. The proposed MAC can be easily extended to support other LPWANs that support time-insensitive applications. The proposed MAC improves the battery lifespan by a maximum of 69.7% and data utility by a maximum of 39% compared to traditional LoRa.

## REFERENCES

- [1] LoRa Alliance, “LoRa alliance,” 2022, <https://lora-alliance.org/>.
- [2] Semtech, “LoRa applications,” 2022, <https://www.semtech.com/lora/lora-applications>.
- [3] “Lifetime of lora devices.” [Online]. Available: <https://info.teledynamics.com/blog/lora-from-the-technology-to-the-application>
- [4] Semtech, “New abi research white paper highlights growth of lora and the lorawan open protocol,” <https://www.semtech.com/company/press/new-abi-research-white-paper-highlights-growth-of-lora-and-the-lorawan-open-protocol>, 2021.
- [5] Soracom Team, “Charging toward sustainability: Solutions to reduce iot device battery waste,” <https://www.soracom.io/blog/charging-toward-sustainability-solutions-to-reduce-iot-device-battery-waste/>, 2022.
- [6] J. Hester and J. Sorber, “The future of sensing is batteryless, intermittent, and awesome,” in *Proceedings of the 15th ACM conference on embedded network sensor systems*, 2017, pp. 1–6.
- [7] A. Swaminathan, “The massive environmental cost of batteries,” <https://www.onio.com/article/environmental-cost-of-batteries.html>, 2020.
- [8] PowerFilm Solar, “Solar power for industrial IoT,” <https://www.powerfilmsolar.com/about-us/the-horizon-blog/2018/07/06/solar-power-for-industrial-iot>, 2020.
- [9] Ilika, “Low powerIoT ecosystem,” <https://www.ilika.com/about-ilika/low-power-iot-ecosystem>, 2020.
- [10] Revibe Energy, “Powering the industrial IoT through vibrations,” 2020. [Online]. Available: <https://revibeenergy.com>

- [11] Balena Blog, “Build a simple solar-powered weather station with lora & the things stack,” <https://www.balena.io/blog/build-a-simple-solar-powered-weather-station-with-lora-the-things-network/>, 2021.
- [12] N. Jackson, J. Adkins, and P. Dutta, “Capacity over capacitance for reliable energy harvesting sensors,” in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, 2019, pp. 193–204.
- [13] B. Xu, A. Oudalov, A. Ulbig, G. Andersson, and D. S. Kirschen, “Modeling of lithium-ion battery degradation for cell life assessment,” *IEEE Transactions on Smart Grid*, 2016.
- [14] N. Maluf, “What happens after 80%?” <https://qnovo.com/what-happens-after-80-percent/>, 2014.
- [15] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, “Towards energy-fairness in lora networks,” in *ICDCS*, 2019.
- [16] SIGFOX, “SIGFOX, The oG Network,” 2022, <http://sigfox.com/en>.
- [17] A. Badam, R. Chandra, J. Dutra, A. Ferrese, S. Hodges, P. Hu, J. Meinershagen, T. Moscibroda, B. Priyantha, and E. Skiani, “Software defined batteries,” in *Proceedings of the 25th Symposium on Operating Systems Principles*, 2015.
- [18] NS-3, “Network simulator,” 2022, <https://www.nsnam.org/>.
- [19] Semtech, “Platform for iot - lora,” <https://www.semtech.com/lora>, 2022.
- [20] S. Fahmida, V. P. Modekurthy, M. Rahman, A. Saifullah, and M. Brocanelli, “Long-lived lora: Prolonging the lifetime of a lora network,” in *ICNP*, 2020.
- [21] L. alliance, “LoRaWAN ADR,” <https://lora-alliance.org/resource-hub/lorawan-specification-v11>, 2017.
- [22] F. A. Kraemer, D. Ammar, A. E. Braten, N. Tamkittikhun, and D. Palma, “Solar energy prediction for constrained iot nodes based on public weather forecasts,” in *ACM International Conference Proceeding Series*, 2017.
- [23] Semtech, “SX1276 datasheet,” 2019. [Online]. Available: <https://semtech.my.salesforce.com/sfc/p/#E000000JelG/a/2R0000001OKx/JUYM3TvBMenQzU4LS8ZJeM58B1jCcoZcUpHV0gnZ.y0>
- [24] U. K. Das, K. S. Tey, M. Seyedmahmoudian, S. Mekhilef, M. Y. I. Idris, V. Van Deventer, B. Horan, and A. Stojcevski, “Forecasting of photovoltaic power generation and model optimization: A review,” *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 912–928, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1364032117311620>
- [25] D. Magrin, M. Centenaro, and L. Vangelista, “Performance evaluation of lora networks in a smart city scenario,” in *ICC*. IEEE, 2017.
- [26] “Solar power data for integration studies,” <https://www.nrel.gov/grid/solar-power-data.html>, 2021.
- [27] Dragino, “Raspberry pi hat featuring gps and lora technology,” <https://www.dragino.com/products/lora/item/106-lora-gps-hat.html>, 2020.
- [28] “Raspberry pi 3 b+,” <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>, 2020.
- [29] “Lmic 1.6 for rpi gps/lora hat,” 2020. [Online]. Available: <https://github.com/wklenk/lmic-rpi-lora-gps-hat>
- [30] “Rak2245 hat/pilot gateway,” 2020. [Online]. Available: [https://cdn-shop.adafruit.com/product-files/4284/4284\\_Get\\_Start\\_with\\_RAK2245\\_Pi\\_HAT\\_V2.4R.pdf](https://cdn-shop.adafruit.com/product-files/4284/4284_Get_Start_with_RAK2245_Pi_HAT_V2.4R.pdf)
- [31] “Chirpstack lorawan network server,” 2020. [Online]. Available: <https://www.chirpstack.io>
- [32] “psutil package,” 2023, <https://pypi.org/project/psutil/>.
- [33] X. Xia, Y. Zheng, and T. Gu, “Ftrack: Parallel decoding for lora transmissions,” in *SenSys*, 2019.
- [34] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, “Improving reliability and scalability of lorawans through lightweight scheduling,” *IEEE Internet of Things Journal*, 2018.
- [35] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, “Scalability analysis of large-scale lorawan networks in ns-3,” *IEEE Internet of Things Journal*, 2017.
- [36] S. Bashash, S. J. Moura, J. C. Forman, and H. K. Fathy, “Plug-in hybrid electric vehicle charge pattern optimization for energy cost and battery longevity,” *Journal of power sources*, 2011.
- [37] L. He, Y.-C. Tung, and K. G. Shin, “icharge: User-interactive charging of mobile devices,” in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, 2017.
- [38] J. Kwak, K. Lee, T. Kim, J. Lee, and I. Shin, “Battery aging deceleration for power-consuming real-time systems,” in *RTSS*. IEEE, 2019.
- [39] A. I. Petriariu, A. Lavric, E. Coca, and V. Popa, “Hybrid power management system for lora communication using renewable energy,” *IEEE Internet of Things Journal*, 2020.